METHOD AND APPARATUS FOR PROVIDING LOCAL DATA PERSISTENCE FOR WEB APPLICATIONS

NOTICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a method and apparatus for providing local data persistence for Web applications.

2. Description of the Related Art

Applications based upon markup languages such as HTML (Hypertext Markup Language) are notoriously well known in the art. In a typical configuration, a user at a client node accesses an HTML document at a server node by having an HTML client application at the client node issue a request to an HTML server application at the server node. Upon receiving such a request, the HTML server at the server node retrieves the requested document and transmits it to the HTML client at the client node. Typically, the client application is a Web browser on the user's personal workstation, while the server application is a Web server at a distant node. Typically, too, the client and the server communicate with each other over a network such as the Internet using a communication protocol such as Transmission Control Protocol/Internet Protocol (TCP/IP). In addition to containing text or graphics for display, an HTML document may contain areas for entry of data by the user, ultimately to be processed on either the client or the server.

A Web application (i.e., an HTML page containing functionality for user data entry) containing potentially secure information needs data persistence to avoid losing the user's data between invocations. One current industry solution for retaining data is JavaScript "cookies", defined in *Teach Yourself Java Script in a Week* (copyright 1996 by Sam.net Publishing) as "a method of storing information locally in the browser and sending it to the server whenever the appropriate pages are requested by the user". Cookies, however, have significant limitations for Web applications that must store data on the client side. Cookies are limited in size (4096 bytes) and the number of entries per domain (20 per cookie file). Also, cookies are not secure because other sites access the same cookie file.

A typical industry solution to storing potentially large amounts of potentially secure data is to use a Common Gateway Interface (CGI) on the server and store the data on a database maintained by the server. This has limitations and complications. The Web site administrator must maintain a list of usernames and passwords to provide security to the individual files. If Secure Sockets Layer (SSL) is not implemented, this is not a very secure method because the data must be transported across the Web while not encrypted.

Another approach (for Microsoft Internet Explorer only) has been a behavior called "userData". This is a function that can save the data in a proprietary format on the computer for retrieval at a later date. However, this solution has several pitfalls. The data is no longer portable, since if the user saves a page on his or her work computer, he or she cannot transport this data to a home computer for later retrieval. Also, this approach is limited to use with the Microsoft Internet Explorer 5.x Web browser, and cannot be used with other browsers such as Netscape Navigator.

SUMMARY OF THE INVENTION

The present invention provides a method for preserving program state data across invocations of a Web browser without the use of cookies, and with the additional benefit of giving users direct control over the disposition of their data.

In accordance with the invention, a Web application dynamically creates a new Web page containing a script function that, when loaded, restores all of the current data to the application. The dynamically created page is then saved locally by the user, using the standard File/Save As function of the client application. Upon return to the Web application, the user is prompted for the location of the saved file. When that location is entered, the page is automatically loaded, the script function run, and the application is returned to the state in which it was left.

As is well known to those skilled in the art, scripting languages such as JavaScript are interpreted language that is used to generate scripts in HTML files that are delimited by <SCRIPT> and </SCRIPT> tags. When an HTML browser encounters such a script in an HTML document that it is processing, the browser executes the statements contained in the script. The present invention uses script functions in the HTML documents making up a Web application to perform the desired operations of saving, restoring and the like.

More particularly, after a user completes a portion or all of the tasks in task panels, he or she can then choose to save the data to a location accessible from the workstation (e.g., a diskette, a zip disk, local hard drive, or a network drive). The data is saved in a file that is generated using the JavaScript interpreter function of the browser. The saved file is an HTML file containing a JavaScript restoration function and the field values the user entered, which are embedded in the script commands. The file is saved using a method similar to that for saving a file in an ordinary client application. The user utilizes the Web browser's ability to perform a "Save As" operation. This saves the dynamically created HTML file. Upon reentry into the tool at some later date the user is prompted for this file. If it exists the user may load the data from the saved file into the tool. This file may be transported via any normal file transfer method (e.g., a diskette, File Transfer Protocol (FTP), etc.) and used at other workstations using the supported browser and having a connection to the Internet Web server that holds the application.

All data gathered stays on the client running the Web browser unless the user explicitly creates the data file and moves it via methods mentioned above. The user has complete control over where their data resides and what is done with it.

The present invention avoids the limitations of the prior art discussed above. The problem of data persistence is solved by saving the data locally. The 4096-byte size limit of cookies is eliminated; the only practical size limit is the memory limit for script functions such as JavaScript (which is almost limitless). The security issues are solved because the user has full control over the location of the data; the user may put it on a removable disk and store it in a secure location (such as a locked desk) if he or she wants. No other Web page has any access to local files, so the security fears of cookies are eliminated. There is no limit on the number of files that a user can save from a particular domain; the user can save as many different files as he or she chooses to and use them on another computer if desired.

Another advantage of the present invention is that to run the Web application, one need not depend on constant or speedy access to the Web. Since the present invention is not in any way tied to the server, it can be used in a disconnected mode from the Web server, and the whole package can be zipped up, downloaded and run locally on any computer by simply pointing a browser at it. Basically, this allows the whole of the application to reside on a disk which is locally accessible to the workstation, and the user can still save the data into the data file. That is to say, the application might reside on a CD and the user may use a laptop to run the tool. All on-line security or performance concerns are instantly eliminated because of this. To the knowledge of the inventors, this functionality has not been previously possible without actually installing an application on the client.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a connected network configuration in which the present invention may be used.

Fig. 2 shows a disconnected "island" configuration in which the present invention may be used.

Fig. 3 shows a data entry window that is displayed in one embodiment of the present invention..

Fig. 4 shows a "save" window that is displayed when the user actuates the save button of the data entry window shown in Fig. 3.

Fig. 5 shows a "load" window that is displayed when the user actuates the load button of the data entry window shown in Fig. 3.

Fig. 6 shows the structure of the source version of the data entry window shown in Fig. 3.

Fig. 7 shows the structure of the source version of the save window shown in Fig. 4.

Fig. 8 shows the structure of the source version of the load window shown in Fig. 5.

Fig. 9 shows the interrelationship among the various documents collectively making up the Web application of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 1 shows one embodiment of the present invention in a network or connected configuration. As shown in the figure, a client/server system 100 comprises a client node 102 coupled to a server node 104 via a network connection 106.

Client node 102 may comprise any suitable personal workstation, such as an Intel-architecture machine running a version of the Microsoft Windows operating system (not separately shown); however, the platform choice is not critical. Executing on client node 102 is an HTML client 108 of any suitable type, such as Netscape Navigator or Microsoft Internet Explorer Web browser. Client node 102 also contains persistent storage for storing client data (in this instance, client-managed pages) 110 in a manner to be described.

HTML client 108 is JavaScript-enabled in the sense that not only can it read and display ordinary HTML files, but it can also interpret any JavaScript function contained in an HTML file being processed.

Similarly, server node 104 may comprise any suitable server system, such as an IBM or Sun Microsystems server, running any suitable operating system (not separately shown) such as Linux or some other version of the UNIX operating system. Executing on server node 104 is an HTML server 112 of any suitable type, such as the Apache Web server or the like. Server node 104 stores server data 114 comprising one or more Web pages thereon, comprising text or graphics, embedded applications, and the like.

Network connection 106 may comprise a public network such as the Internet, a private network such as a corporate intranet, or a network combining certain aspects of public and private networks, such as a virtual private network (VPN). The network connection 106 may be implemented in any suitable manner, such as a dialup connection, a local area network (LAN) connected to a wide area network (WAN) via a gateway, or the like.

In a manner well known in the art, as described in such publications as L. Aronson and J. Lowery, HTML 3.2 Manual of Style (1997), at pages 1-7, a user at the client node 102 accesses a Web page at the server node 104 by having the HTML client 108 issue a request over the network connection 106 containing the Internet address of the server node 104 and the directory location of the page being requested. Upon receiving such a request over the network connection 106, the HTML server 112 at server node 104 retrieves the requested page and forwards it to the HTML client at client node 102.

In the system 100 shown in Fig. 1, the HTML client 108 and the HTML server 112 reside at different nodes. While this is the customary configuration, it is not the only possible configuration. For example, one may dispense with the server node 104 and manage all data, including Web pages ordinarily stored at the server node, as client data 110 in a disconnected "island" configuration 200 as shown in Fig. 2.

A Web application that implements this invention would typically function as follows. When a user wants to save data for a current session with the application, he or she clicks a "Save" button. The application dynamically creates a new file. This file contains a JavaScript function that is loaded whenever the user returns to the browser session. Once this page is created, the user is prompted to save the page locally. The user must manually perform this step because JavaScript itself is restricted from directly accessing the user's hard drive.

On returning to the application, the user is prompted for the location of the saved file. The user enters the location, and the application loads the file and runs its function. The application resumes at the state in which the user left it, with all previous data and at the same point in the process.

Fig. 3 shows a data entry 300 window that is displayed to the user of the HTML client 108 in one embodiment of the present invention. Data entry window 300 comprises a header frame 300a and a base frame 300b. Header 302a is a persistent frame that is used while a session is active to hold (but not display) data that is entered by the user. Base frame 302b is the frame where the data is gathered from the user and where all output is displayed. The user traverses the Web pages within the application 900 using this frame.

As shown in the figure, the base frame 300b of data entry window 300 contains a data entry area 302 as well as a "save" button 304, a "load" button 306 and a "build" button 308. The user navigates to various lines of the data entry area 302 to enter data (in this case, sysplex configuration data). When the user wants to save previously entered data, he or she actuates the save button 304 (as by clicking on it with a mouse), resulting in the display of a "save" window 400 shown in Fig. 4. Similarly, if the user wants to load previously saved data, he or she actuates the load button 306, resulting in the display of a "load" window 500 shown in Fig. 5. Finally, when the user has finished entering data, he or she may actuate a "build" button 308 to have the data processed by the Web application.

Fig. 4 shows the save window 400 that is displayed when the user actuates the save button 304 in the data entry window 300. Save window 400 contains a text message 402 that prompts the user to save the previously entered data locally, such as by selecting "File" on the toolbar 404, then selecting "Save As" on the drop-down menu that then appears, and entering the requested filename and path information. The "Save As" function is a standard function of HTML clients 108 such as Netscape Navigator and Microsoft Internet Explorer and therefore will not be described further in this specification.

Fig. 5 shows the load window 500 that is displayed when the user actuates the load button 306 in the data entry window 300. Load window 500 displays a text message 502 requesting the user to enter the name of the saved file in an area 504, possibly with the assistance of a "browse" button 506 if the user does not remember the filename or if it is stored in a different directory. The user then clicks on a "Reload Previous Data" link 508 to reload the previously saved data.

Fig. 9 shows the interrelationship among the various documents collectively making up the Web application 900 of the present invention. These include a frameset document 902, a header document 904, a base frame document 906, and a load document 908. Each of these documents is a separate HTML file. Frameset document 902 contains a head 910 that contains a script portion 912 and a body 914 that contains a first frame portion 916 and a second frame portion 918. Frame portions 916 and 918 define the boundaries of the displayed frames 300a and 300b of data entry window and contain references 920 and 922 to header file 904 and to base frame file 906, respectively, which contain the actual content of the frames. In addition to generating the display of the header 300a, header file 904 holds all the application data that is entered by the user in a form named "holddata" and in various arrays. Header file 904 only holds this data while the Web application is open and the user is interacting with it; for persistent storage of user data, the method of the present invention is used.

Fig. 6 shows the general structure of the base frame document 906 that is displayed as the base frame 300b of the data entry window 300. As shown in the figure, the base frame document 906 comprises a script portion 602 and a form portion 604. Script portion 602 in turn contains a

JavaScript function 606 for opening the load window 500. Form portion 604 contains, among other elements for eliciting user data, a build button portion 608 from which build button 308 is generated, a load button portion 610 from which load button 306 is generated, and a save button portion 612 from which save button 304 is generated. JavaScript function 606 is invoked when the user actuates the load button 306, as indicated by the line 614. In a similar manner, the JavaScript function in the frameset document 902 for saving user data is invoked when the user actuates the save button 306, as indicated by the line 616.

Fig. 7 shows the general structure of the HTML source file 700 (alternatively, the "save" file or the "save" page) that is displayed as the save window 400. In contrast to the preexisting documents 902-908 that make up the Web application 900, save file 700 is dynamically generated by the script function 912 in the head portion 910 of frameset document 902 when the user actuates the save button 304 in the data entry window 300. As shown in the figure, the source file 700 contains a form portion 702 that contains a table portion 704 and a script portion 706. Table portion 704 contains an HTML encoding of the displayed text message 402, while script portion 706 contains a script function for restoring the saved data to the data entry page 300. Script portion 706 is invoked when the file 700 is loaded from the client data area 110 on user actuation of the load link 508.

Fig. 8 shows the general structure of the load page 908 that is displayed as the load window 500. As shown in the figure, load page 908 contains a form portion 802 that generates the displayed elements 502-508 and a script portion 804 that opens the save file 700 containing the locally saved user data in response to user actuation of the reload link 508 in load window 500.

Appendix A shows the JavaScript function saveToDisk() that is invoked when the user actuates the save button 304 in window 300. This function, which the resides in script portion 912 of frameset document 902, dynamically creates the save file 700 (Fig. 7), which is saved by the user locally (e.g., to disk) as an HTML file in user data 110. More particularly, as shown in this listing, the function saveToDisk() opens a new window (line 4), writes individual lines of data from the header document 904 to the HTML file 700 defining the window (lines 6-48), and then closes the data stream to allow the newly generated window to be displayed as window 400 (line 49). Upon being reopened, this HTML file 700 repopulates the header document 904 of the Web application 900 with previously entered data, thus providing multi-session data persistence for a client-side Web application.

In Appendix A, the terms listed below have the following meaning:

'arrayList[]': An array in the header file 904 that contains the names of all other arrays in the file. This provides a way to add more arrays in the application without modifying this function to look for each individual array.

'url': Page currently on when this function is called (xxxxx.html). This provides a way to renter the application at the point of saving.

'toolDescription': One-line text description of tool ("My Web Application")

Appendix B is a listing of other JavaScript functions in the script portion 912 of the frameset document 902. Lines 51-55 of this listing contain the JavaScript function set(field,newdata), which repopulates fields of the header document 904 with new data from the save document 700 when the save document 700 is loaded. This function is invoked by the JavaScript function saveFields() in the save document 700 through its statements of the form opener.parent.set(. . .), shown as lines 339-477 in Appendix D. Lines 60-66 of this listing contain the JavaScript function get(name), which retrieves the value of the field whose name is supplied as an input.

Appendix C is an HTML source listing of the base frame document 906 shown in Figs. 6 and 9. Lines 79-42 of this listing (delimited by the tags <SCRIPT . . .> and </SCRIPT>) contain script portion 602, which in turn contains the script function 606, loadFromDisk(), for opening load document 908 (lines 117-123). Lines 150-262 of this listing contain form portion 604 (delimited by the tags <FORM . . .> and </FORM>), which in turn contains build button portion 608 (line 254), load button portion 610 and save button portion 612 (line 256).

As noted above, and as indicated by the onClick="loadFromDisk();" attribute in line 256, actuation of the load button invokes the JavaScript function loadFromDisk() on lines 117-123. On the other hand, as indicated by the onClick=parent.saveToDisk("interviews/ps_topics.html") attribute in line 256, actuation of the save button invokes the JavaScript function saveToDisk() that is contained in the script portion 912 of the frameset document 902 and is reproduced in Appendix A.

Appendix D is an HTML source listing of the save file 700 shown in Fig. 7. Lines 521-526 of this listing (delimited by the tags <FORM . . .> and </FORM>) contain the form portion 702 Form portion 702 contains the table portion 704 (lines 303-332) for generating the displayed text message, as well as the script portion 706 (lines 336-496) delimited by the tags <SCRIPT . . .> and </SCRIPT> for restoring the saved data to the data entry page 300. Script portion 706 contains the script function saveFields() (lines 338-493), which is invoked when the page is loaded, as indicated by the onLoad="saveFields" attribute in line 301. On being invoked, the script function saveFields() repopulates header document 904, corresponding to the header portion 300a of the data entry page 300, with the previously saved data. As can be seen from scanning the lines of the function saveFields(), the data to be restored is embedded in the script function itself as the second argument of the various lines.

Appendix E is an HTML source listing of the load document 908 shown in Figs. 8 and 9. Lines 521-526 of this listing (delimited by the tags <FORM . . .> and </FORM>) contain the form portion 802 for generating displayed elements 502-508. Lines 511-516 of this listing (delimited by the tags <SCRIPT> and </SCRIPT>) contain the script portion 804 for opening the saved page 400. The JavaScript function setlink() (lines 512-515) contained in script portion 804 is invoked when the user clicks on the reload link 508 (line 525), as indicated by the onClick=setlink() attribute in line 522.

Appendix F is an HTML listing for the header document 904 shown in Fig. 9.

While a particular embodiment has been shown and described, various modifications will be apparent to those skilled in the art. Thus, while the Web documents generated in the preferred embodiment use the HTML markup language, the invention is not restricted to any particular markup language. Also, while the Web documents generated in the preferred embodiment use JavaScript functions to provide the desired functionality, other scripting languages supported by the Web browser could be used instead.

Appendix A: Function saveToDisk() in Frameset Document
Copyright © 1999 IBM Corporation

```
function saveToDisk(url) {

var xdist = screen.availWidth - 643;
remote =
window.open('',"saveWin","WIDTH=631,HEIGHT=350,screeny=0,screenx=" +
xdist + ",resizable=0,menubar=1,status=0,scrollbars=0");
remote.opener.name = "opener";
remote.document.write('<HTML>\n<HEAD>\n');
remote.document.write('<SCRIPT LANGUAGE="JavaScript1.2">\n');
remote.document.write('function saveFields(){\n');
remote.document.write('}\n </SCR' + 'IPT>\n');
remote.document.write('\n</HEAD>\n');
remote.document.write('<BODY onLoad="saveFields()" onBlur="var timerID =
setTimeout(\'window.close()\', 20000)">\n<FORM NAME="restore">\n');
remote.document.write('<TABLE BORDER="0" WIDTH="560">\n');
remote.document.write('<TR>\n<TD>\n');
remote.document.write('<TABLE BORDER="0" WIDTH="100%">\n<TR
BGCOLOR="#c8d8f8">\n<TD>\n<FONT SIZE="2" FACE="Arial, helv">\n');
remote.document.write('<B>' + toolDescription + ': Save & Restore
Data</B>\n</FONT>\n</TD>\n</TR>\n</TABLE>\n</TD>\n</TR>\n');
remote.document.write('<TR><TD><P>\n This document contains the
information you have entered into the ' + toolDescription + '.');
remote.document.write('</P></TD></TR>\n');
remote.document.write('<TR><TD><TABLE WIDTH="100%" BORDER="0">\n<TR>');
remote.document.write('<TD VALIGN=Top><FONT COLOR="blue"><STRONG>To Save
Your Data:</STRONG></FONT>\n <OL>');
remote.document.write('<LI>Select <STRONG>File</STRONG> and then
<STRONG>Save as...</STRONG> from the menubar at the top of this
window.</LI>\n');
remote.document.write('<LI>In the Save As... dialog box, choose a name
and directory location for the configuration data file and click the
Save button.</LI>\n');
remote.document.write('<LI>Close this
Window.</LI>\n</OL></TD>\n</TR>\n');
remote.document.write('<TR><TD><HR NOSHADE COLOR="#CCCCCC">Once you have
saved your data, you can quit the ' + toolDescription + ' and return to
it later. ');
remote.document.write('When you return, you will be given instructions
to load the saved file and resume your work.</TD></TR>')
remote.document.write('</TABLE></TD></TR>\n');
remote.document.write('<TR>\n<TD WIDTH="100%" HEIGHT="40">\n');
remote.document.write('<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3"
WIDTH="100%">\n<TR><TD COLSPAN="6" WIDTH="100%"><HR NOSHADE
COLOR="#CCCCCC"></TD></TR>\n');
remote.document.write('</TABLE>\n</TD></TR></TABLE>\n');
remote.document.write('<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><3R>\n');
remote.document.write('<A NAME="loading"><HR></A><BR><FONT SIZE=5><STRONG>Loading Data...</STRONG></FONT>\n');
remote.document.write('<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>\n');
remote.document.write('&LT;SCRIPT LANGUAGE="JavaScript1.2"&GT;\n');
remote.document.write('function saveFields(){\n');
for (var i = 0; i < get("holddatalength"); i++) {
  remote.document.write("opener.parent.set('" + elemname(i) + "','" + get(elemname(i)) + "');\n");
  }
remote.document.write('\n');
remote.document.write('<SCRIPT LANGUAGE="JavaScript1.2">\n');
for (i = 0; i < header.arrayList.length; i++) {
remote.document.write('var ' + header.arrayList[i] + ' = ');
remote.document.write('opener.header.' + header.arrayList[i] + '\n');
remote.document.write('document.write("opener.parent.header.' + header.arrayList[i] + ' = " + ' + header.arrayList[i] + ' + ";\\n");');
}//end for
remote.document.write('\n </SCR' + 'IPT>\n');
remote.document.write('var timeID = setTimeout("opener.document.location.reload(); window.close()", 1500)\n');
remote.document.write('}');
remote.document.write('\n &LT;/SCRIPT&GT;\n');
remote.document.write('</FORM></BODY>\n</HTML>');
remote.document.close();
}
```

Appendix B: Other Functions in Frameset Document
Copyright © 1999 IBM Corporation

```
function set(field,newdata)
 {
  doSet = "header.document.holddata." + field + ".value = " + '"""' + " +
'" + newdata + "';";
  eval(doSet);
 }
// get receives one input and returns one value.
// The input is the name of the field whose value is to be returned.
// This function reads text from the fields of the HOLDDATA form and
returns it
// to the calling program.
function get(name)
 { if (name == "holddatalength") {return
header.document.holddata.length}
    else { doGet = "ret = header.document.holddata." + name + ".value";
    eval(doGet);
    return ret
    }
 }
```

```
<BASE HREF="http://www.s390.ibm.com/pso/psotool/interviews/">
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso-8859-1">
  <META NAME="owner"     CONTENT="gdunlap@us.ibm.com">
  <META NAME="keywords"  CONTENT="S/390 IBM System/390 OS/390 parallel
sysplex configuration setup interactive">
  <META NAME="abstract"  CONTENT="This web-based tool helps S/390
customers migrate to a Parallel Sysplex environment.">
  <META NAME="author"    CONTENT="gcorbin@us.ibm.com">
  <META NAME="contact"   CONTENT="gcorbin@us.ibm.com">
  <META NAME="review"    CONTENT="19991231">
  <META NAME="security"  CONTENT="public">
  <SCRIPT language="JavaScript1.1">

    // basic configuration status flags
    naming_status = parent.get("naming_status");
    naming_valid = "valid";
    if ((parent.get("names1_valid") ==
"invalid")||(parent.get("names2_valid") == "invalid")) {naming_valid =
"invalid"}

    sw_status = parent.get("sw_status");
    sw_valid = parent.get("sw1_valid");

    hw_status = parent.get("hw_status");
    hw_valid = "valid";
    if ((parent.get("hw3_valid") == "invalid")||(parent.get("hw5_valid")
== "invalid")) {hw_valid = "invalid"}

    str1_status = parent.get("str1_status");
    str1_valid = parent.get("str1_valid");

    cds_status = parent.get("cds_status");
    cds_valid = "valid";
    if ((parent.get("cds1_valid") ==
"invalid")||(parent.get("cds_sfm_valid") ==
"invalid")||(parent.get("cds_arm_valid") ==
"invalid")||(parent.get("cds_wlm_valid") ==
"invalid")||(parent.get("cds_logr_valid") == "invalid"))
      {cds_valid = "invalid"}

    sfm_status = parent.get("sfm_status");
    sfm_valid = "valid";
    jes2_status = parent.get("jes2_status");
    jes2_valid = parent.get("jes2_valid");
    racf_status = parent.get("racf_status");
    racf_valid = "valid";
    tape_status = parent.get("tape_status");
    tape_valid = "valid";
    ecs_status = parent.get("ecs_status");
    ecs_valid = "valid";
    logrec_status = parent.get("logrec_status");
    logrec_valid = parent.get("logrec_valid");
    operlog_status = parent.get("operlog_status");
    operlog_valid = parent.get("operlog_valid");
    parent.set("URL","interviews/ps_topics.html");

function loadFromDisk()
  {
   var xdist = screen.availWidth - 455;
   load =
window.open('../load.html','loadWin','WIDTH=400,HEIGHT=105,screeny=405,s
creenx=' + xdist + ',resizable=0,menubar=0,status=0,scrollbars=1');
   load.opener.name = 'opener';
   load.focus();
  }

function parentLink(url) {
parent.opener.document.location=url
parent.opener.focus();
}
  function buildCheck(url)
   {
    // first see if all required tasks have been completed
    if ((naming_status == "Not done")||(sw_status == "Not
done")||(hw_status == "Not done")||(str1_status == "Not
done")||(cds_status == "Not done"))
       {alert("Please complete all required tasks before continuing with
the Build step.")}
    else {
          if ((naming_valid == "invalid")||(sw_valid ==
"invalid")||(hw_valid == "invalid")||(str1_valid ==
"invalid")||(cds_valid == "invalid")||(sfm_valid ==
"invalid")||(jes2_valid == "invalid")||(racf_valid ==
"invalid")||(tape_valid == "invalid")||(ecs_valid ==
"invalid")||(logrec_valid == "invalid")||(operlog_valid == "invalid"))
                    {alert("Please correct all invalid entries before
continuing with the Build step.")}
```

```
            else{window.location = url}
          }
   } // end function


</SCRIPT>
  <LINK REL="STYLESHEET" TYPE="text/css" HREF="/include/text.css">
</HEAD>

<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER="0" WIDTH="560">
<TR>
 <TD>
   <FORM NAME="interview1">
   <TABLE BORDER="0" WIDTH="100%">
      <TR BGCOLOR="#c8d8f8">
        <TD>
          <FONT SIZE="2" FACE="Arial, helv">
          <B>Parallel Sysplex Configuration: Interviews</B>
          </FONT>
        </TD>
        <TD align="right">
          <FONT SIZE="2" FACE="Arial, helv">
          <A
HREF="javascript:parentLink('../ps_intro.html');">Introduction</A>
          </FONT>
        </TD>
      </TR>
    </TABLE>
  </TD>
</TR>
<TR>
  <TD>
    <FONT COLOR="#000000" SIZE="2" FACE="Arial, helv">
     We begin with a series of interviews in which you'll answer
questions
     about the sysplex configuration that you are creating. Complete
each interview topic
     that is marked <B>Required</B>. When you have finished answering
all of
     the interview questions, click <B>Build</B>. The Parallel Sysplex
     Configuration Assistant will build a checklist of steps for you to
follow,
     as well as customized jobs and other data sets for you to use.
     </FONT>
  </TD>
```

```
</TR>
<TR><TD><IMG SRC="../images/complete.gif"><STRONG> = Complete
</STRONG>&nbsp;&nbsp;&nbsp;
        <IMG SRC="../images/alert.gif"><STRONG> = Invalid
Data</STRONG></TD></TR>
<TR>
  <TD>
     <FORM = "interview1">
     <TABLE BORDER="1" CELLPADDING=2 WIDTH="100%">
        <TR VALIGN="center" BGCOLOR="#C8D8F8">
           <TH WIDTH="*"><FONT SIZE="2" FACE="Helvetica,
Arial">Interview Topics: Sysplex Definition</FONT></TH>
           <TH WIDTH="20"><FONT SIZE="2" FACE="Helvetica,
Arial">Required</FONT></TH>
           <TH WIDTH="20"><FONT SIZE="2" FACE="Helvetica,
Arial">Status</FONT></TH>
        </TR>
        <TR VALIGN="center">
           <TD><A HREF="ps_names1.html">Sysplex-wide naming
conventions</A></TD>
           <TD ALIGN="center">Yes</TD>
           <TD CELLPADDING=0>&nbsp;<SCRIPT>if (naming_status == "Done")
{document.write('<IMG SRC="../images/complete.gif">')}
                                            if (naming_valid == "invalid")
{document.write('<IMG SRC="../images/alert.gif">')}</SCRIPT></TD>
        </TR>
        <TR VALIGN="center">
           <TD><A HREF="ps_sw1.html">Software environment</A></TD>
           <TD ALIGN="center">Yes</TD>
           <TD CELLPADDING=0>&nbsp;<SCRIPT>if (sw_status == "Done")
{document.write('<IMG SRC="../images/complete.gif">')}
                                            if (sw_valid == "invalid")
{document.write('<IMG SRC="../images/alert.gif">')}</SCRIPT></TD>
        </TR>
        <TR VALIGN="center">
           <TD><A HREF="ps_hw2.html">Hardware components</A></TD>
           <TD ALIGN="center">Yes</TD>
           <TD CELLPADDING=0>&nbsp;<SCRIPT>if (hw_status == "Done")
{document.write('<IMG SRC="../images/complete.gif">')}
                                            if (hw_valid == "invalid")
{document.write('<IMG SRC="../images/alert.gif">')}</SCRIPT></TD>
        </TR>
       <TR VALIGN="center">
          <TD><A HREF="ps_cds1.html">Couple data sets</A></TD>
          <TD ALIGN="center">Yes</TD>
          <TD CELLPADDING=0>&nbsp;<SCRIPT>if (cds_status == "Done")
{document.write('<IMG SRC="../images/complete.gif">')}
```

```
                                        if (cds_valid == "invalid") {document.write('<IMG SRC="../images/alert.gif">')}</SCRIPT></TD>
      </TR>
      <TR VALIGN="center">
        <TD><A HREF="ps_sfm_pols.html">Sysplex failure management (SFM) policies</A></TD>
        <TD ALIGN="center">No</TD>
        <TD CELLPADDING=0>&nbsp;<SCRIPT>if (sfm_status == "Done") {document.write('<IMG SRC="../images/complete.gif">')}
                                        if (sfm_valid == "invalid") {document.write('<IMG SRC="../images/alert.gif">')}</SCRIPT></TD>
      </TR>
       <TR VALIGN="center" BGCOLOR="#C8D8F8">
         <TH WIDTH="*"><FONT SIZE="2" FACE="Helvetica, Arial">Interview Topics: Resource Sharing</FONT></TH>
         <TH WIDTH="20"><FONT SIZE="2" FACE="Helvetica, Arial">Required?</FONT></TH>
         <TH WIDTH="20"><FONT SIZE="2" FACE="Helvetica, Arial">Status</FONT></TH>
       </TR>
<TR VALIGN='center'><TD><A HREF='ps_jes2.html'>JES2 checkpoint data</A></TD><TD ALIGN='center'>No</TD><TD CELLPADDING=0> </TD></TR><TR VALIGN='center'><TD><A HREF='ps_racf.html'>OS/390 Security Server database</A></TD><TD ALIGN='center'>No</TD><TD CELLPADDING=0> </TD></TR>
      <TR VALIGN="center">
        <TD><A HREF="ps_tape.html">Tape devices (automatic tape sharing feature)</A></TD>
        <TD ALIGN="center">No</TD>
        <TD CELLPADDING=0> </TD>
      </TR>
      <TR VALIGN="center">
        <TD><A HREF="ps_ecs.html">Catalogs (enhanced catalog sharing feature)</A></TD>
        <TD ALIGN="center">No</TD>
        <TD CELLPADDING=0> </TD>
      </TR>
      <TR VALIGN="center">
        <TD><A HREF="ps_operlog.html">OPERLOG (system logger feature)</A></TD>
        <TD ALIGN="center">No</TD>
        <TD CELLPADDING=0> </TD>
      </TR>
      <TR VALIGN="center">
        <TD><A HREF="ps_logrec.html">LOGREC (system logger feature)</A></TD>
        <TD ALIGN="center">No</TD>
        <TD CELLPADDING=0> </TD>
```

```
            </TR>
            <TR VALIGN="center">
               <TD><A HREF="ps_str_map.html">Coupling Facility structure
mapping</A></TD>
               <TD ALIGN="center">Yes</TD>
               <TD CELLPADDING=0> </TD>
            </TR>
           <TR VALIGN="center">
                <TD CELLPADDING=0 COLSPAN=3>
                     <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0
WIDTH=100%><TR VALIGN="center">
                     <TD ALIGN="left"><INPUT TYPE="button" NAME=Build
VALUE="Build" onClick="buildCheck('../build/bld_basic.html');"></TD>
                     <TD ALIGN="right">
             <INPUT TYPE="button" NAME=Load VALUE="Load Configuration Data"
onClick="loadFromDisk();"><INPUT TYPE="button" NAME=Save VALUE="Save
Configuration Data"
onClick=parent.saveToDisk("interviews/ps_topics.html")>
             </TD>
                     </TR></TABLE>
                </TD>
          </TR>
        </TABLE>
        </FORM>
    </TD>
</TR>
</TABLE>
</WhiteSpace>
</TD>
</TR></TABLE>
</TR></TABLE>
</TR></TABLE>

<TABLE width="600" border="0" cellspacing="0" cellpadding="0">
<TR bgcolor="#000000">
<TD align="center" width="49"><A href="http://www.ibm.com/privacy/"
class="nav" style="color: #ffffff;"><A
href="http://www.ibm.com/privacy/" class="nav" Ttyle="color:
#ffffff;"><FONT face="Arial, sans-serif" size="-2"
color="#ffffff"><B>Privacy</B></FONT></A></TD>
<TD bgcolor="#959595" width="1"><IMG src="http://www.ibm.com/i/c.gif"
width="1" height="21"/></TD>
<TD align="center" width="49"><A href="http://www.ibm.com/legal/"
class="nav" style="color: #ffffff;"><A href="http://www.ibm.com/legal/"
class="nav" Ttyle="color: #ffffff;"><FONT face="Arial, sans-serif"
size="-2" color="#ffffff"><B>Legal</B></FONT></A></TD>
```

```
<TD bgcolor="#959595" width="1·"><IMG src="http://www.ibm.com/i/c.gif"
width="1" height="1"/></TD>
<TD align="center" width="49"><A href="http://www.ibm.com/contact/"
class="nav" style="color: #ffffff;"><A
href="http://www.ibm.com/contact/" class="nav" Ttyle="color:
#ffffff;"><FONT face="Arial, sans-serif" size="-2"
color="#ffffff"><B>Contact</B></FONT></A></TD>
<TD bgcolor="#959595" width="1"><IMG src="http://www.ibm.com/i/c.gif"
width="1" height="1"/></TD>
<TD width="450"> </TD></TR>
</TABLE>


</BODY>

</BODY>
</HTML>
```

Appendix D: Source Version of Save Page

```
<BASE HREF="http://www.s390.ibm.com/pso/psotool/">
<HTML>
<HEAD>
<META NAME="abstract" CONTENT="IBM System/390">
<META NAME="keywords" CONTENT="S/390 Parallel Sysplex Configuration Assistant">
<META NAME="owner"    CONTENT="gcorbin@us.ibm.com">
<META NAME="author"   CONTENT="George Corbin">
<META NAME="review" CONTENT="990928">
<META NAME="security" CONTENT="public">
<TITLE>S/390 Parallel Sysplex Configuration Assistant (Save & Restore Data)</TITLE>


</HEAD>
<BODY onLoad="saveFields()" onBlur="var timerID = setTimeout('window.close()', 20000)">
<FORM NAME="restore">
<TABLE BORDER="0" WIDTH="560">
<TR>
<TD>
<TABLE BORDER="0" WIDTH="100%">
<TR BGCOLOR="#c8d8f8">
<TD>
<FONT SIZE="2" FACE="Arial, helv">
<B>S/390 Parallel Sysplex Configuration Assistant: Save & Restore Data</B>
</FONT>
</TD>
</TR>
</TABLE>
</TD>
</TR>
<TR><TD><P>
 This document contains the information you have entered into the S/390 Parallel Sysplex Configuration Assistant.</P></TD></TR>
<TR><TD><TABLE WIDTH="100%" BORDER="0">
<TR><TD VALIGN=Top><FONT COLOR="blue"><STRONG>To Save Your Data:</STRONG></FONT>
 <OL><LI>Select <STRONG>File</STRONG> and then <STRONG>Save as...</STRONG> from the menubar at the top of this window.</LI>
<LI>In the Save As... dialog box, choose a name and directory location for the configuration data file and click the Save button.</LI>
<LI>Close this Window.</LI>
```

```
</OL></TD>
</TR>
<TR><TD><HR NOSHADE COLOR="#CCCCCC">Once you have saved your data, you can quit the S/390 Parallel Sysplex Configuration Assistant and return to it later. When you return, you will be given instructions to load the saved file and resume your work.</TD></TR></TABLE></TD></TR>
<TR>
<TD WIDTH="100%" HEIGHT="40">
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3" WIDTH="100%">
<TR><TD COLSPAN="6" WIDTH="100%"><HR NOSHADE COLOR="#CCCCCC"></TD></TR>
</TABLE>
</TD></TR></TABLE>
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>
<A NAME="loading"><HR></A><BR><FONT SIZE=5><STRONG>Loading Data...</STRONG></FONT>
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>
<SCRIPT LANGUAGE="JavaScript1.2">

function saveFields(){
opener.parent.set('hassaved','yes');
opener.parent.set('URL','interviews/ps_topics.html');
opener.parent.set('netscape_browser','');
opener.parent.set('naming_status','Not done');
opener.parent.set('sw_status','Not done');
opener.parent.set('hw_status','Not done');
opener.parent.set('str1_status','Not done');
opener.parent.set('cds_status','Not done');
opener.parent.set('sfm_status','Not done');
opener.parent.set('jes2_status','Not done');
opener.parent.set('racf_status','Not done');
opener.parent.set('tape_status','Not done');
opener.parent.set('ecs_status','Not done');
opener.parent.set('operlog_status','Not done');
opener.parent.set('logrec_status','Not done');
opener.parent.set('names1_valid','valid');
opener.parent.set('names2_valid','valid');
opener.parent.set('sw1_valid','valid');
opener.parent.set('hw3_valid','valid');
opener.parent.set('hw5_valid','valid');
opener.parent.set('cds1_valid','valid');
opener.parent.set('cds_sfm_valid','valid');
opener.parent.set('cds_arm_valid','valid');
opener.parent.set('cds_wlm_valid','valid');
opener.parent.set('cds_logr_valid','valid');
opener.parent.set('jes2_valid','valid');
opener.parent.set('operlog_valid','valid');
opener.parent.set('logrec_valid','valid');
opener.parent.set('strl_valid','valid');
opener.parent.set('ps_con_ae_valid','invalid');
opener.parent.set('formatds_status','yes');
opener.parent.set('hlq','');
opener.parent.set('hlq2','');
opener.parent.set('suffix','PS');
opener.parent.set('pvolser','');
opener.parent.set('avolser','');
opener.parent.set('bvolser','');
opener.parent.set('plexname','SYSPLEX1');
opener.parent.set('memnum','2');
opener.parent.set('maxmem','8');
opener.parent.set('sysaction','  ');
opener.parent.set('sys_select','  ');
opener.parent.set('cfnum','2');
opener.parent.set('cfaction','  ');
opener.parent.set('cf_select','  ');
opener.parent.set('sub_num','10');
opener.parent.set('con_action','  ');
opener.parent.set('con_select','  ');
opener.parent.set('time_source','etr');
opener.parent.set('etrzone','yes');
opener.parent.set('simetrid','01');
opener.parent.set('gmt_direction','west');
opener.parent.set('gmt_hours','00');
opener.parent.set('gmt_minutes','00');
opener.parent.set('jesver','JES2');
opener.parent.set('secprod','RACF');
opener.parent.set('grsprod','GRS');
opener.parent.set('sms_active','yes');
opener.parent.set('cics','yes');
opener.parent.set('cics_regions','');
opener.parent.set('ims','yes');
opener.parent.set('dfsmshsm','yes');
opener.parent.set('rmm','yes');
opener.parent.set('dbshared','no');
opener.parent.set('racfdb_ds','1');
opener.parent.set('racf_b_size','1024');
opener.parent.set('jes_mas','yes');
opener.parent.set('jes_strl','CKPT1');
opener.parent.set('jesdsn1','');
opener.parent.set('jesdsn2','');
opener.parent.set('jesaltds','');
opener.parent.set('jesvol1','');
opener.parent.set('jesvol2','');
opener.parent.set('jesaltvol','');
opener.parent.set('jes_records','1500');
opener.parent.set('plexdsn1','');
opener.parent.set('plexdsn2','');
opener.parent.set('plexdsn3','');
opener.parent.set('xcfgroup','100');
opener.parent.set('plex_pvolser','');
opener.parent.set('plex_avolser','');
opener.parent.set('plex_bvolser','');
opener.parent.set('cfrmdsn1','');
opener.parent.set('cfrmdsn2','');
opener.parent.set('cfrmdsn3','');
opener.parent.set('cfrm_pvolser','');
opener.parent.set('cfrm_avolser','');
opener.parent.set('cfrm_bvolser','');
opener.parent.set('sfmdsn1','');
opener.parent.set('sfmdsn2','');
opener.parent.set('sfmdsn3','');
opener.parent.set('sfm_pvolser','');
opener.parent.set('sfm_avolser','');
opener.parent.set('sfm_bvolser','');
opener.parent.set('sfm_maxpol','6');
opener.parent.set('sfmpols','1');
opener.parent.set('polaction','  ');
opener.parent.set('polselect','  ');
opener.parent.set('armdsn1','');
opener.parent.set('armdsn2','');
opener.parent.set('armdsn3','');
opener.parent.set('arm_pvolser','');
opener.parent.set('arm_avolser','');
opener.parent.set('arm_bvolser','');
opener.parent.set('arm_maxpol','6');
opener.parent.set('armpols','1');
opener.parent.set('armaction','  ');
opener.parent.set('armselect','  ');
opener.parent.set('wlmdsn1','');
opener.parent.set('wlmdsn2','');
opener.parent.set('wlmdsn3','');
opener.parent.set('wlm_pvolser','');
opener.parent.set('wlm_avolser','');
opener.parent.set('wlm_bvolser','');
opener.parent.set('wlm_maxpol','6');
opener.parent.set('wkloads','32');
opener.parent.set('srvclass','128');
opener.parent.set('applenv','100');
opener.parent.set('schenv','100');
opener.parent.set('logrdsn1','');
opener.parent.set('logrdsn2','');
opener.parent.set('logrdsn3','');
opener.parent.set('logr_pvolser','');
opener.parent.set('logr_avolser','');
opener.parent.set('logr_bvolser','');
opener.parent.set('logrec_records','');
opener.parent.set('logrec_hlq','');
opener.parent.set('logrec_ucat','');
opener.parent.set('logrec_vol','');
opener.parent.set('oplog_records','');
opener.parent.set('oplog_hlq','');
opener.parent.set('oplog_ucat','');
opener.parent.set('oplog_vol','');
opener.parent.set('oplog_cf_time','30');
opener.parent.set('oplog_lib','');
opener.parent.set('oplog_dasd_time','7 ');
opener.parent.set('tapenum','');
opener.parent.set('maxcat','1');
opener.parent.set('xcfmem','50');

opener.parent.header.sysArray = [["SYS1", "CPC1", "LP1", [["NORMAL",
1]], [], [], [], []], ["SYS2", "CPC2", "LP2", [["NORMAL", 1]], [], [],
[], []]];
opener.parent.header.cfArray = [["CF1", "9674", "02", "000000000000",
"00", "1", "6000"], ["CF2", "9674", "02", "000000000000", "00", "1",
"6000"]];
opener.parent.header.sfmArray = [["NORMAL", "All systems have equal
weight."]];
opener.parent.header.armArray = [["ARMPOL00", "Unless an element is
specifically defined in an ARM policy, it will fall into this restart
group."]];
opener.parent.header.conArray = [["PLEXMSTR", "", "3270-X", "MASTER",
"", "", "*ALL"]];
opener.parent.header.strArray = [["XCF", "IXCPATH1", "956", "956", 0,
1], ["XCF", "IXCPATH2", "16316", "16316", 1, 0], ["GRS", "ISGLOCK",
"8448", "8448", 0, 1]];
opener.parent.header.grsDSName = [];
opener.parent.header.grsHLName = [];
opener.parent.header.checklist = [];
opener.parent.header.threshStructure = ["IXCPATH1", "IXCPATH2", "ISGLOCK", "CKPT1", "IRRXCF00_P", "IRRXCF00_B", "IEFAUTOS", "SYSIGGCAS_ECS", "OPERLOG", "LOGREC"];
opener.parent.header.threshValue = ["80", "80", "80", "80", "95", "95", "90", "80", "90", "90"];
opener.parent.header.rel9 = [];

var timeID = setTimeout("opener.document.location.reload(); window.close()", 1500)
}


</SCRIPT>
</FORM></BODY>
</HTML>
```

Appendix E: Source Version of Load Page

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
  <META NAME="owner"     CONTENT="jim@us.ibm.com">
  <META NAME="keywords"  CONTENT="S/390 IBM System/390 OS/390 parallel sysplex configuration setup interactive">
  <META NAME="abstract"  CONTENT="This web-based tool helps S/390 customers migrate to a Parallel Sysplex environment.">
  <META NAME="author"    CONTENT="gcorbin@us.ibm.com">
  <META NAME="contact"   CONTENT="gcorbin@us.ibm.com">
  <META NAME="review"    CONTENT="19991231">
  <META NAME="security"  CONTENT="public">
  <TITLE>Parallel Sysplex Configuration Tool: Introduction and Task Menu</TITLE>
<SCRIPT>
function setlink(){
var newurl = "file:///" + document.findfile.getfile.value + "#loading";
document.links[0].href=newurl;
}
</SCRIPT>

</HEAD>

<BODY bgcolor="#FFFFFF">
<FORM NAME="findfile">
<P><FONT COLOR="#000000" size="2" face="Arial, helv, helvetica, sans serif"><STRONG>To Reload Previous Data:<BR>1) Enter Filename:</STRONG>
<input type=file name=getfile value="" onClick=setlink(); onBlur=setlink();><BR>
<STRONG>2) And Click: <BR></STRONG></FONT>
<A HREF="file:///C|/sysplexg.html#loading">Reload Previous Data</A></P>
</FORM>
</BODY>
</HTML>
```

```
<BASE HREF="http://www.s390.ibm.com/pso/psotool/">
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso-8859-1">
  <META NAME="owner"     CONTENT="gdunlap@us.ibm.com">
  <META NAME="keywords"  CONTENT="S/390 IBM System/390 OS/390 parallel
sysplex configuration setup interactive">
  <META NAME="abstract"  CONTENT="This web-based tool helps S/390
customers migrate to a Parallel Sysplex environment.">
  <META NAME="author"    CONTENT="gcorbin@us.ibm.com">
  <META NAME="contact"   CONTENT="gcorbin@us.ibm.com">
  <META NAME="review"    CONTENT="19991231">
  <META NAME="security"  CONTENT="public">

  <LINK REL="STYLESHEET" TYPE="text/css" HREF="/include/text.css">
</HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">

function redirect()
{
for (var i=0; i<document.links.length; i++)
    {
     document.links[i].href= "javascript:parentLink('" +
document.links[i].href + "')";
    }
}

function parentLink(url) {
  parent.opener.document.location=url;
  parent.opener.focus();
}


</SCRIPT>
<BODY BGCOLOR="white" alink="white" vlink="white" topmargin="0"
leftmargin="0" marginheight="0" marginwidth="0"
onLoad="parent.getCookieData(); redirect();
parent.basefrm.location='interviews/ps_topics.html';">
<table width="100%" cellspacing="0" cellpadding="0" border="0">
<tr><td align="left" BGCOLOR="#0033CC" valign="top"><A
HREF="http://www.ibm.com/" border="0" target="new"><img
src="http://www.ibm.com/i/v9/m/en/ibm_logo.gif" WIDTH=57  height=24
alt="IBM" border="0" align="left"></A><img
```

```
src="http://www.ibm.com/i/v9/m/en/logo_sp.gif" width="100%" height=24
alt="" border="0" align="left"></td>
</table>




<SCRIPT LANGUAGE="JavaScript1.2">

// set recieves two inputs: a field name and a value to change that
field to.
// This function writes values to the hidden text fields in the HOLDDATA
form on this page.
function set(field,newdata)
 {
  doSet = "document.holddata." + field + ".value = " + '""' + " + " + '" +
newdata + "';";
  eval(doSet);
 }
// get receives one input and returns one value.
// The input is the name of the field whose value is to be returned.
// This function reads text from the fields of the HOLDDATA form and
returns it
// to the calling program.
function get(name)
 { if (name == "holddatalength") {return document.holddata.length}
   else { doGet = "ret = document.holddata." + name + ".value";
  eval(doGet);
  return ret}
 }
function name(index)
 { return document.holddata[index].name }

</SCRIPT>

<FORM NAME="holddata">
<!-- Variable indicating if data has been saved -->
<INPUT TYPE="hidden" NAME="hassaved"  VALUE="no">
<INPUT TYPE="hidden" NAME="URL"    VALUE="interviews/ps_topics.html">
<INPUT TYPE="hidden" NAME="netscape_browser"  VALUE="">

<!-- planning task status indicators for basic setup -->
<INPUT TYPE="hidden" NAME="naming_status"  VALUE="Not done">
<INPUT TYPE="hidden" NAME="sw_status"      VALUE="Not done">
<INPUT TYPE="hidden" NAME="hw_status"      VALUE="Not done">
<INPUT TYPE="hidden" NAME="str1_status"    VALUE="Not done">
<INPUT TYPE="hidden" NAME="cds_status"     VALUE="Not done">
```

```
<INPUT TYPE="hidden" NAME="sfm_status"     VALUE="Not done">
<!-- planning task status indicators for resource sharing setup -->
<INPUT TYPE="hidden" NAME="jes2_status"     VALUE="Not done">
<INPUT TYPE="hidden" NAME="racf_status"     VALUE="Not done">
<INPUT TYPE="hidden" NAME="tape_status"     VALUE="Not done">
<INPUT TYPE="hidden" NAME="ecs_status"      VALUE="Not done">
<INPUT TYPE="hidden" NAME="operlog_status"  VALUE="Not done">
<INPUT TYPE="hidden" NAME="logrec_status"   VALUE="Not done">

<!-- validation placeholders for each page -->
<INPUT TYPE="hidden" NAME="names1_valid"    VALUE="valid">
<INPUT TYPE="hidden" NAME="names2_valid"    VALUE="valid">
<INPUT TYPE="hidden" NAME="sw1_valid"       VALUE="valid">
<INPUT TYPE="hidden" NAME="hw3_valid"       VALUE="valid">
<INPUT TYPE="hidden" NAME="hw5_valid"       VALUE="valid">
<INPUT TYPE="hidden" NAME="cds1_valid"      VALUE="valid">
<INPUT TYPE="hidden" NAME="cds_sfm_valid"   VALUE="valid">
<INPUT TYPE="hidden" NAME="cds_arm_valid"   VALUE="valid">
<INPUT TYPE="hidden" NAME="cds_wlm_valid"   VALUE="valid">
<INPUT TYPE="hidden" NAME="cds_logr_valid"  VALUE="valid">
<INPUT TYPE="hidden" NAME="jes2_valid"      VALUE="valid">
<INPUT TYPE="hidden" NAME="operlog_valid"   VALUE="valid">
<INPUT TYPE="hidden" NAME="logrec_valid"    VALUE="valid">
<INPUT TYPE="hidden" NAME="str1_valid"      VALUE="valid">
<INPUT TYPE="hidden" NAME="ps_con_ae_valid" VALUE="invalid">


<!-- build status indicators for basic setup-->
<INPUT TYPE="hidden" NAME="formatds_status"  VALUE="yes">

<!-- naming conventions -->
<INPUT TYPE="hidden" NAME="hlq"            VALUE="">
<INPUT TYPE="hidden" NAME="hlq2"           VALUE="">
<INPUT TYPE="hidden" NAME="suffix"         VALUE="PS">
<INPUT TYPE="hidden" NAME="pvolser"        VALUE="">
<INPUT TYPE="hidden" NAME="avolser"        VALUE="">
<INPUT TYPE="hidden" NAME="bvolser"        VALUE="">

<!-- hardware environment -->
<INPUT TYPE="hidden" NAME="plexname"       VALUE="SYSPLEX1">
<INPUT TYPE="hidden" NAME="memnum"         VALUE="2">
<INPUT TYPE="hidden" NAME="maxmem"         VALUE="8">
<INPUT TYPE="hidden" NAME="sysaction">
<INPUT TYPE="hidden" NAME="sys_select">
<INPUT TYPE="hidden" NAME="cfnum"          VALUE="2">
```

```
<INPUT TYPE="hidden" NAME="cfaction">
<INPUT TYPE="hidden" NAME="cf_select">
<INPUT TYPE="hidden" NAME="sub_num"          VALUE="10">
<INPUT TYPE="hidden" NAME="con_action">
<INPUT TYPE="hidden" NAME="con_select">
<INPUT TYPE="hidden" NAME="time_source"      VALUE="etr">
<INPUT TYPE="hidden" NAME="etrzone"          VALUE="yes">
<INPUT TYPE="hidden" NAME="simetrid"         VALUE="01">
<INPUT TYPE="hidden" NAME="gmt_direction"    VALUE="west">
<INPUT TYPE="hidden" NAME="gmt_hours"        VALUE="00">
<INPUT TYPE="hidden" NAME="gmt_minutes"      VALUE="00">

<!-- software environment -->
<INPUT TYPE="hidden" NAME="jesver"           VALUE="JES2">
<INPUT TYPE="hidden" NAME="secprod"          VALUE="RACF">
<INPUT TYPE="hidden" NAME="grsprod"          VALUE="GRS">
<INPUT TYPE="hidden" NAME="sms_active"       VALUE="yes">
<INPUT TYPE="hidden" NAME="cics"             VALUE="yes">
<INPUT TYPE="hidden" NAME="cics_regions"     VALUE="">
<INPUT TYPE="hidden" NAME="ims"              VALUE="yes">
<INPUT TYPE="hidden" NAME="dfsmshsm"         VALUE="yes">
<INPUT TYPE="hidden" NAME="rmm"              VALUE="yes">

<!-- security stuff -->
<INPUT TYPE="hidden" NAME="dbshared"         VALUE="no">
<INPUT TYPE="hidden" NAME="racfdb_ds"        VALUE="1"
<INPUT TYPE="hidden" NAME="racf_p_size"      VALUE="5120">
<INPUT TYPE="hidden" NAME="racf_b_size"      VALUE="1024">

<!-- jes2 stuff -->
<INPUT TYPE="hidden" NAME="jes_mas"          VALUE="yes">
<INPUT TYPE="hidden" NAME="jes_str1" SIZE="8" VALUE="CKPT1">
<INPUT TYPE="hidden" NAME="jesdsn1"          VALUE="">
<INPUT TYPE="hidden" NAME="jesdsn2"          VALUE="">
<INPUT TYPE="hidden" NAME="jesaltds"         VALUE="">
<INPUT TYPE="hidden" NAME="jesvol1" SIZE="6" VALUE="">
<INPUT TYPE="hidden" NAME="jesvol2" SIZE="6" VALUE="">
<INPUT TYPE="hidden" NAME="jesaltvol" SIZE="6" VALUE="">
<INPUT TYPE="hidden" NAME="jes_records"      VALUE="1500">


<!-- Sysplex couple dataset stuff -->
<INPUT TYPE="hidden" NAME="plexdsn1"         VALUE="">
<INPUT TYPE="hidden" NAME="plexdsn2"         VALUE="">
<INPUT TYPE="hidden" NAME="plexdsn3"         VALUE="">
```

```
<INPUT TYPE="hidden" NAME="xcfgroup"      VALUE="100">
<INPUT TYPE="hidden" NAME="plex_pvolser"  VALUE="">
<INPUT TYPE="hidden" NAME="plex_avolser"  VALUE="">
<INPUT TYPE="hidden" NAME="plex_bvolser"  VALUE="">

<!-- CFRM couple dataset stuff -->
<INPUT TYPE="hidden" NAME="cfrmdsn1"      VALUE="">
<INPUT TYPE="hidden" NAME="cfrmdsn2"      VALUE="">
<INPUT TYPE="hidden" NAME="cfrmdsn3"      VALUE="">
<INPUT TYPE="hidden" NAME="cfrm_pvolser"  VALUE="">
<INPUT TYPE="hidden" NAME="cfrm_avolser"  VALUE="">
<INPUT TYPE="hidden" NAME="cfrm_bvolser"  VALUE="">

<!-- sfm stuff -->
<INPUT TYPE="hidden" NAME="sfmdsn1"       VALUE="">
<INPUT TYPE="hidden" NAME="sfmdsn2"       VALUE="">
<INPUT TYPE="hidden" NAME="sfmdsn3"       VALUE="">
<INPUT TYPE="hidden" NAME="sfm_pvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="sfm_avolser"   VALUE="">
<INPUT TYPE="hidden" NAME="sfm_bvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="sfm_maxpol"    VALUE="6">
<INPUT TYPE="hidden" NAME="sfmpols"       VALUE="1">
<INPUT TYPE="hidden" NAME="polaction">
<INPUT TYPE="hidden" NAME="polselect">

<!-- arm stuff -->
<INPUT TYPE="hidden" NAME="armdsn1"       VALUE="">
<INPUT TYPE="hidden" NAME="armdsn2"       VALUE="">
<INPUT TYPE="hidden" NAME="armdsn3"       VALUE="">
<INPUT TYPE="hidden" NAME="arm_pvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="arm_avolser"   VALUE="">
<INPUT TYPE="hidden" NAME="arm_bvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="arm_maxpol"    VALUE="6">
<INPUT TYPE="hidden" NAME="armpols"       VALUE="1">
<INPUT TYPE="hidden" NAME="armaction">
<INPUT TYPE="hidden" NAME="armselect">

<!-- wlm stuff -->
<INPUT TYPE="hidden" NAME="wlmdsn1"       VALUE="">
<INPUT TYPE="hidden" NAME="wlmdsn2"       VALUE="">
<INPUT TYPE="hidden" NAME="wlmdsn3"       VALUE="">
<INPUT TYPE="hidden" NAME="wlm_pvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="wlm_avolser"   VALUE="">
<INPUT TYPE="hidden" NAME="wlm_bvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="wlm_maxpol"    VALUE="6">
```

```
<INPUT TYPE="hidden" NAME="wkloads"        VALUE="32">
<INPUT TYPE="hidden" NAME="srvclass"       VALUE="128">
<INPUT TYPE="hidden" NAME="applenv"        VALUE="100">
<INPUT TYPE="hidden" NAME="schenv"         VALUE="100">

<!-- logger stuff -->
<INPUT TYPE="hidden" NAME="logrdsn1"       VALUE="">
<INPUT TYPE="hidden" NAME="logrdsn2"       VALUE="">
<INPUT TYPE="hidden" NAME="logrdsn3"       VALUE="">
<INPUT TYPE="hidden" NAME="logr_pvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="logr_avolser"   VALUE="">
<INPUT TYPE="hidden" NAME="logr_bvolser"   VALUE="">
<INPUT TYPE="hidden" NAME="logrec_records" VALUE="">
<INPUT TYPE="hidden" NAME="logrec_hlq"     VALUE="">
<INPUT TYPE="hidden" NAME="logrec_ucat"    VALUE="">
<INPUT TYPE="hidden" NAME="logrec_vol"    VALUE="">
<INPUT TYPE="hidden" NAME="oplog_records"  VALUE="">
<INPUT TYPE="hidden" NAME="oplog_hlq"      VALUE="">
<INPUT TYPE="hidden" NAME="oplog_ucat"     VALUE="">
<INPUT TYPE="hidden" NAME="oplog_vol"     VALUE="">
<INPUT TYPE="hidden" NAME="oplog_cf_time"  VALUE="30">
<INPUT TYPE="hidden" NAME="oplog_lib"      VALUE="">
<INPUT TYPE="hidden" NAME="oplog_dasd_time"  VALUE="7 ">

<!-- automatic tape sharing stuff -->
<INPUT TYPE="hidden" NAME="tapenum"        VALUE="">

<!-- ECS stuff -->
<INPUT TYPE="hidden" NAME="maxcat"         VALUE="1">

<!-- BUILD it -->
<INPUT TYPE="hidden" NAME="xcfmem"        VALUE="50">
</FORM>
<SCRIPT LANGUAGE="JavaScript1.2">




//---------------------------------------------------------------------------------------------------//
// This is an array to keep track of the arrays.  If you add an array, please add it's name here.  //
arrayList = new Array("sysArray","cfArray","sfmArray","armArray","conArray","strArray","grsDSName","grsHLName","checklist","threshStructure","threshValue","rel9");
```

```
//-------------------------------------------------------------------------------------------------//

// This section builds an initial sysArray, the array of systems in the sysplex
        sysCols = 3
        sysArray = new Array();
        memnum = 2;
        for (rowCnt=0; rowCnt < memnum; rowCnt++)
        {
          sysArray[rowCnt] = new Array();
          sysArray[rowCnt][0] = "SYS" + (rowCnt+1);
          sysArray[rowCnt][1] = "CPC" + (rowCnt+1);
          sysArray[rowCnt][2] = "LP" + (rowCnt+1);
// SFM related data
          sysArray[rowCnt][3] = new Array();
          sysArray[rowCnt][3][0] = new Array();
          sysArray[rowCnt][3][0][0] = "NORMAL";
          sysArray[rowCnt][3][0][1] = 1;
// MCS-related data
          sysArray[rowCnt][4] = new Array();  // which consoles it physically connects to
          sysArray[rowCnt][5] = new Array();  // which consoles it listens to for commands
          sysArray[rowCnt][6] = new Array();  // which consoles it routes messages to
          sysArray[rowCnt][7] = new Array();  // os390 release
        } //end for loop on rowCnt
 // This section builds an initial cfArray, the array of Coupling Facilities in the sysplex
        cfCols = 7;
        cfArray = new Array();
        cfnum = 2;
        for (rowCnt=0; rowCnt < cfnum; rowCnt++)
        {
          cfArray[rowCnt] = new Array();
          cfArray[rowCnt][0] = "CF" + (rowCnt+1);
          cfArray[rowCnt][1] = "9674";
          cfArray[rowCnt][2] = "02";
          cfArray[rowCnt][3] = "000000000000";
          cfArray[rowCnt][4] = "00";
          cfArray[rowCnt][5] = "1";
          cfArray[rowCnt][6] = "6000";
         } // end of loop on rowCnt
 // This section builds an initial sfmArray, the array of SFM policies
```

```
          sfmArray = new Array();
          // Policy 1, Normal
             sfmArray[0] = new Array();
             sfmArray[0][0] = "NORMAL";
             sfmArray[0][1] = "All systems have equal weight.";
    // This section builds an initial armArray, the array of ARM policies
          armArray = new Array();
          // Build row 0 -- entries for sample policy ARMPOL00
          armArray[0] = new Array();
          armArray[0][0] = "ARMPOL00";
          armArray[0][1] = "Unless an element is specifically defined in an
   ARM policy, it will fall into this restart group.";
    // This section builds an initial conArray, the array of MCS consoles
          conArray = new Array();
          // a sample console definition entry
             conArray[0] = new Array();
             conArray[0][0] = "PLEXMSTR";        // console name
             conArray[0][1] = "";               // device number
             conArray[0][2] = "3270-X";         // unit type
             conArray[0][3] = "MASTER";         // authority
             conArray[0][4] = "";               // route code
             conArray[0][5] = "";           // Connected to field
             conArray[0][6] = "*ALL";           // Connected to field
    // This section builds an initial strArray, the array of CF structures
    // Start with columns 0 thru 3 for component name (e.g., GRS),
   structure name, initial size, and max size
          strArray = new Array();
          // Build rows 0 thru 3 -- entries for XCF structures
          strArray[0] = new Array();
          strArray[0][0] = "XCF";
          strArray[0][1] = "IXCPATH1";
          strArray[0][2] = "956";
          strArray[0][3] = "956";
          strArray[0][4] = 0;
          strArray[0][5] = 1;

          strArray[1] = new Array();
          strArray[1][0] = "XCF";
          strArray[1][1] = "IXCPATH2";
          strArray[1][2] = "16316";
          strArray[1][3] = "16316";
          strArray[1][4] = 1;
          strArray[1][5] = 0;

          // Build row 2 -- entry for GRS structure
          strArray[2] = new Array();
    strArray[2][0] = "GRS";
    strArray[2][1] = "ISGLOCK";
    strArray[2][2] = "8448";
    strArray[2][3] = "8448";
    strArray[2][4] = 0;
    strArray[2][5] = 1;

    // Continue with columns 4 thru n for CF preferences
    cfnum = 2;
    for (rowCnt=0;rowCnt < 5; rowCnt++)
     {
       for (colCnt=4; colCnt < cfnum; colCnt++)
        {
           strArray[rowCnt][colCnt] = "";
           } // end colCnt loop
     } // end rowCnt loop

     // array for hlding DataSet names for GRS insertions into
GRSRNLxx Parmlib member
     // Input is received from the SW section panel 2
     grsDSName = new Array();
     grsHLName = new Array();
     // array for storing status on build page
     checklist = new Array();

     threshStructure = new Array();
     threshStructure[0] = "IXCPATH1";
     threshStructure[1] = "IXCPATH2";
     threshStructure[2] = "ISGLOCK";
     threshStructure[3] = "CKPT1";
     threshStructure[4] = "IRRXCF00_P";
     threshStructure[5] = "IRRXCF00_B";
     threshStructure[6] = "IEFAUTOS";
     threshStructure[7] = "SYSIGGCAS_ECS";
     threshStructure[8] = "OPERLOG";
     threshStructure[9] = "LOGREC";

     threshValue = new Array();
     threshValue[0] = "80";
     threshValue[1] = "80";
     threshValue[2] = "80";
     threshValue[3] = "80";
     threshValue[4] = "95";
     threshValue[5] = "95";
     threshValue[6] = "90";
```

```
        threshValue[7] = "80";
        threshValue[8] = "90";
        threshValue[9] = "90";

        rel9 = new Array();



</SCRIPT>
</WhiteSpace>
</td>
</tr></table>
</tr></table>
</tr></table>

<table width="600" border="0" cellspacing="0" cellpadding="0">
<tr bgcolor="#000000">
<td align="center" width="49"><a href="http://www.ibm.com/privacy/"
class="nav" style="color: #ffffff;"><font face="Arial, sans-serif"
size="-2" color="#ffffff"><b>Privacy</b></font></a></td>
<td bgcolor="#959595" width="1"><img src="http://www.ibm.com/i/c.gif"
width="1" height="21"/></td>
<td align="center" width="49"><a href="http://www.ibm.com/legal/"
class="nav" style="color: #ffffff;"><font face="Arial, sans-serif"
size="-2" color="#ffffff"><b>Legal</b></font></a></td>
<td bgcolor="#959595" width="1"><img src="http://www.ibm.com/i/c.gif"
width="1" height="1"/></td>
<td align="center" width="49"><a href="http://www.ibm.com/contact/"
class="nav" style="color: #ffffff;"><font face="Arial, sans-serif"
size="-2" color="#ffffff"><b>Contact</b></font></a></td>
<td bgcolor="#959595" width="1"><img src="http://www.ibm.com/i/c.gif"
width="1" height="1"/></td>
<td width="450">&nbsp;</td></tr>
</table>
<SCRIPT LANGUAGE="JavaScript">

<!C- Keylime Software  4.1 13/07/2000
// IBM Baseline Tag
var kl_siteID = "77";
var kl_tagProtocol = "";
var kl_akamaipath = "a1944.g.akamai.net/f/1944/1482/8h/";
if (location.protocol == "

</BODY>

</BODY>
```

```
</HTML>
```